**Intel® AI DLA MŁODZIEŻY 2019**

PROGRAM PILOTAŻOWY

# 

# PODRĘCZNIK MODERATORA

**Moduł 14 - Pozyskiwanie [Koncepcje typowe dla poszczególnych domen] (AI dla komputerowego rozpoznawania obrazów)**

CAŁKOWITY CZAS TRWANIA SESJI: 480 MINUT

LICZBA INSTRUKTORÓW: 2

LICZBA UCZESTNIKÓW: 40

**UWAGA:** ŻADNA OSOBA SPOZA ORGANIZACJI INTEL CORPORATION NIE MOŻE KOPIOWAĆ, ROZPROWADZAĆ, REPRODUKOWAĆ ANI W ŻADEN INNY SPOSÓB UDOSTĘPNIAĆ INFORMACJI ZAWARTYCH W NINIEJSZYM DOKUMENCIE OSOBOM, INSTYTUCJOM I/LUB ORGANIZACJOM NIEPOWIĄZANYM Z NINIEJSZYM PROGRAMEM BEZ ZGODY WYŻEJ WYMIENIONEJ ORGANIZACJI.

NIEPRZESTRZEGANIE POWYŻSZYCH ZASAD BĘDZIE SKUTKOWAĆ ZAKOŃCZENIEM WSPÓŁPRACY PRZEZ WYŻEJ WYMIENIONĄ ORGANIZACJĘ.

NINIEJSZY DOKUMENT MUSI **ZOSTAĆ ZWRÓCONY** UPOWAŻNIONEMU PERSONELOWI FIRMY INTEL CORPORATION PO ZAKOŃCZENIU SESJI.

## **Podręcznik moderatora**

|  |  |
| --- | --- |
| **Temat lekcji:** Moduł 14 - Pozyskiwanie [Koncepcje typowe dla poszczególnych domen] (AI dla komputerowego rozpoznawania obrazów) | **Tryb:** Sesje interaktywne prowadzona przez instruktora |
| **Podsumowanie:** Ten moduł prezentuje podstawy teoretyczne kluczowych koncepcji i technik w domenie CV (komputerowe rozpoznawanie obrazów) przed ich szczegółowym omówieniem na etapie Doświadczenie. Omówione koncepcje/techniki obejmują piksele, konwolucyjne sieci neuronowe i różne algorytmy AI związane z CV. | |
| **Cele edukacyjne:**   1. Zrozumienie sposobu interpretacji obrazów przez komputer 2. Przegląd procesu AI w domenie CV 3. Rozwój intuicyjnego rozumienia typowych technik AI w domenie CV 4. Prezentacja potencjalnych zastosowań rozwiązań CV wykorzystujących AI 5. Prezentacja biblioteki OpenCV i korzystanie z jej podstawowych funkcji | |
| **Rezultaty edukacyjne:**   1. Wyjaśnienie sposobu analizy obrazów przez komputery 2. Wyjaśnienie działania konwolucji 3. Opisanie sposobu działania konwolucyjnych sieci neuronowych 4. Opisanie sposobu działania maszyny wektorów nośnych 5. Wskazanie zastosowań komputerowego rozpoznawania obrazów 6. Korzystanie z funkcji biblioteki OpenCV | |
| **Wymagania wstępne:**   1. Znajomość języka Python na potrzeby podstawowej nauki o danych | |
| **Kluczowe pojęcia:**   1. Piksele 2. Matryca 3. Cechy obrazu 4. Maszyny wektorów nośnych 5. Konwolucyjne sieci neuronowe 6. Klasyfikacja, rozpoznawanie, segmentacja, lokalizacja | |
| **Kluczowe umiejętności:**   1. Umiejętności typowe dla domeny CV: Zrozumienie koncepcji i pojęć typowych dla domeny | |
| **Wykorzystywane materiały:**   1. [Slajdy] Moduł 14 - Pozyskiwanie [Koncepcje typowe dla poszczególnych domen] (AI dla komputerowego rozpoznawania obrazów) 2. [Notatnik Jupyter - Uczestnicy] Moduł 14 - Wprowadzenie do OpenCV 3. [Quiz] Moduł 14 - Pozyskiwanie [Koncepcje typowe dla poszczególnych domen] (Komputerowe rozpoznawanie obrazów) | |
| **Kwestie etyczne dotyczące AI:**   1. Deepfake | |
| **Zastosowanie w scenariuszach z prawdziwego życia [np. kwestie społeczne]:**   1. Zastosowania CV, np. kamery bezpieczeństwa, auta samojezdne | |

## 

## **1. Konspekt**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nie** | **Czas** | **Czynność** | **Opis** | **Cel** |
| 1. Wprowadzenie do komputerowego rozpoznawania obrazów | 30 minut | Wykład: Wprowadzenie do komputerowego rozpoznawania obrazów | Wprowadzenie  - Różne rodzaje zastosowania CV  - Proces AI w zastosowaniach CV | Wprowadzenie do technologii komputerowego rozpoznawania obrazów i jej zastosowań |
| 2. Podstawowe pojęcia komputerowego rozpoznawania obrazów | 35 minut | Wykład: Jak komputery widzą obrazy? | Piksele | Znajomość podstawowych pojęć związanych z komputerowym rozpoznawaniem obrazów |
| Ćwiczenie: Jak komputery widzą obrazy? | Ćwiczenie z pikselami: https://www.w3schools.com/colors/colors\_rgb.asp |
| 25 minut | Wykład: Cechy obrazu | Wprowadzenie do cech obrazu |
| 90 minut | Samokształcenie: Wprowadzenie do OpenCV | Wprowadzenie do biblioteki OpenCV z użyciem notatnika Jupyter |
| 5. Znajomość algorytmów uczenia maszynowego (ML): Operator konwolucji | 20 minut | Wykład: Operator konwolucji | Operator konwolucji | Znajomość operatora konwolucji |
| 40 minut | Ćwiczenie: Operator konwolucji | Ćwiczenie: Operator konwolucji <https://www.youtube.com/watch?v=YgtModJ-4cw&t=3s> (film objaśniający + ćwiczenie pisemne) |
| 30 minut | Ćwiczenie: Aplikacji internetowa do konwolucji | - Objaśnienie różnych filtrów (kernel) <http://matlabtricks.com/post-5/3x3-convolution-kernels-with-online-demo>  - Funkcje różnych filtrów  - Tworzenie własnego filtra |
| 20 minut | Quiz | Quiz Kahoot |
| 4. Znajomość algorytmów uczenia maszynowego (ML): Konwolucyjne sieci neuronowe | 40 minut | Wykład: Konwolucyjne sieci neuronowe (CNN) | - Kernel czyli filtr  - Różne warstwy w CNN  - Trenowanie CNN | Intuicyjne zrozumienie konwolucyjnych sieci neuronowych |
| 20 minut | Ćwiczenie: Testowanie sieci konwolucyjnej za pomocą zbioru danych MNIST | Ćwiczenie: Testowanie sieci konwolucyjnej za pomocą zbioru danych MNIST <http://www.denseinl2.com/webcnn/digitdemo.html>  <https://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html>  <http://scs.ryerson.ca/~aharley/vis/conv/flat.html> |
| 20 minut | Ćwiczenie: Maszyna zdolna do nauki | - Trenowanie modelu w praktyce za pomocą aplikacji internetowej <https://experiments.withgoogle.com/teachable-machine> |
| 20 minut | Ćwiczenie: Działanie CNN w praktyce | - Działanie CNN w praktyce: <https://cloud.google.com/vision/> |
| 20 minut | Quiz | Quiz Kahoot |
| 3. Znajomość algorytmów uczenia maszynowego (ML): Maszyna wektorów nośnych (SVM) | 20 minut | Wykład: Wprowadzenie do maszyn wektorów nośnych | - Wektor nośny  - Hiperpłaszczyzny | Intuicyjne zrozumienie maszyn wektorów nośnych |
| 10 minut | Quiz | Quiz Kahoot |
| 4. | 40 minut | Dyskusja na temat etyki | Dyskusja na temat deepfake | Zachęcenie uczestników, by brali pod wagę etyczne aspekty związane z komputerowym rozpoznawaniem obrazów |

**2. Przygotowanie sesji**

**Instalacja:**

1. Skopiuj katalog „**Moduł 14 - Pliki robocze (Uczestnicy)**” z notatnikiem Jupyter na wszystkie laptopy.
2. Laptopy (Intel x64) z kamerą internetową powinny zostać zainstalowane i skonfigurowane z programem Intel® Distribution for Python i biblioteką OpenCV. W tym celu można:
   1. Zainstalować program Anaconda po pobraniu go z witryny <https://www.continuum.io/downloads>
   2. Uruchomić wiersz poleceń Anaconda jako administrator w systemie Windows i wprowadzić następujące polecenia:
      1. conda update conda
      2. conda config –add channels intel
      3. conda create –n idp3 intelpython3\_full python=3
      4. Więcej pomocnych informacji można znaleźć tutaj: <https://software.intel.com/en-us/articles/intel-distribution-for-python-development-environment-setting-for-jupyter-notebook-and>
   3. Zainstalować bibliotekę OpenCV w środowisku idp3. Najprostszym sposobem jest skopiowanie plików „cv2.pyd” i „opencv\_world400.dll” z podkatalogu „[ComputerVision] Acquire /setup” do <Anaconda Directory>/envs/idp3[nazwa twojego środowiska]/Lib/site-packages
   4. Po wykonaniu kroku 2 b. iii. zostanie utworzony folder
   5. Jeśli kopiujesz pliki, pamiętaj, że pobierając, kopiując, instalując lub używając oprogramowania, akceptujesz warunki umowy licencyjnej OpenCV („setup/License.txt”). Więcej informacji znajduje się tutaj: <https://opencv.org/license.html>.
   6. Aby uruchomić notatnik jupyter w środowisku idp3, uruchom wiersz poleceń:
      1. activate idp3
      2. jupyter notebook

(Jest też dostępna instrukcja instalacji na wypadek, gdyby uczestnicy mieli sami zainstalować OpenCV)

## 

## **3. Przewodnik po czynnościach**

***[Slajdy 1–14] Wprowadzenie do komputerowego rozpoznawania obrazów [30 min]***

**Celem tej części jest wprowadzenie do zagadnień związanych z pikselami jako istotnych elementów składowych w obszarze zastosowania AI do komputerowego rozpoznawania obrazów.**

***[Slajd 2]***

Komputerowe rozpoznawanie obrazów polega na tym, że uczymy komputery widzieć.

***[Slajdy 3–5]***

Określenie „komputerowe rozpoznawanie obrazów” (ang. computer vision) powstało stosunkowo niedawno, choć ta technologia ma już dużo praktycznych zastosowań. Na przykład:

* Skanowanie kodów kreskowych
* Czujniki odcisków palców
* Skanowanie D
* Rozpoznawanie twarzy

<https://www.youtube.com/watch?v=eQLcDmfmGB0>

***[Slajd 6]***

Jakie zadania możemy wykonać?

Dla pojedynczego obiektu:

- Klasyfikacja: Ustalenie, co jest obecne na obrazie. Obrazowi zostaje przypisana „etykieta”.

- Lokalizacja: Ustalenie, gdzie na obrazie obiekt się znajduje.

- Klasyfikacja + Lokalizacja: Ustalenie, jaki obiekt jest obecny i gdzie na obrazie ten obiekt się znajduje.

Dla wielu obiektów:

- Wykrywanie obiektów: Identyfikacja wystąpień obiektów ze świata realnego (twarzy, rowerów, budynków itd.) na obrazie lub w nagraniu wideo.

- Segmentacja wystąpienia: Wykrywanie wystąpień, nadawanie kategorii, etykietowanie pikseli

***[Slajd 7]***

Jak wygląda proces AI w przypadku komputerowego rozpoznawania obrazów?

Czy pamiętacie proces AI? Przypomnijmy sobie szablon procesu AI.

***[Slajd 8]***

Podczas tej sesji będziemy używać tego opisu problemu: Jak poprawić bezpieczeństwo za pomocą komputerowego rozpoznawania obrazów?

Postanowiliście, że chcecie, aby komputer rozpoznawał twarze osób uprawnionych do wejścia i wysyłał alarm, gdy ktoś nieuprawniony próbuje wejść.

***[Slajd 9]***

Zaczynacie więc pozyskiwać dane, czyli zdjęcia osób, które są uprawnione. Jak inaczej można pozyskać dane?

***[Slajd 10]***

Nadajecie danym format kompatybilny z waszym modelem.

Jak można ro zrobić?

* Najpierw przeglądacie zdjęcia i wybieracie te, które się nadają. Spójrzcie na zdjęcia. Zauważcie, że są to zbiory zdjęć bardzo różnego rodzaju [użyj zdjęć open source] <http://vis-www.cs.umass.edu/lfw/number_11.html>
* Następnie chcecie, żeby zdjęcia były w odpowiednim rozmiarze. Niektóre zdjęcia trzeba będzie przyciąć albo zmienić ich rozmiar.

***[Slajd 11]***

Macie już zbiory danych gotowe do modelowania! W przypadku komputerowego rozpoznawania obrazów dostępne są liczne modele do różnych zastosowań. Sprawdźcie listę gotowych modeli!

Na tym kursie nie będziemy się zajmować każdym modelem, lecz chcemy wam zaprezentować modele niezbyt skomplikowane, ale też takie, które mogą wam pomóc w projektach!

* Maszyna wektorów nośnych (SVM)
* Konwolucyjna sieć neuronowa (CNN)

Zajmiemy się tym szczegółowo nieco później.

Wybieracie modele, uruchamiacie je, sprawdzacie ich wydajność, a następnie wybieracie ten model, który najlepiej się nadaje do waszego projektu!

***[Slajd 12]***

Później wdrożycie rozwiązanie i przetestujecie je w warunkach rzeczywistych.

***[Slajd 13]***

Tak zatem wygląda cały proces!

***[Slajd 14]***

Oto, czego jeszcze nauczymy się na tej sesji

***[Slajdy 15–21] Wykład: Jak komputery widzą obrazy? [35 min]***

**Celem tej części jest wprowadzenie do zagadnień związanych z pikselami jako istotnych elementów składowych w obszarze zastosowania AI do komputerowego rozpoznawania obrazów.**

***[Slajd 16]***

Czy zastanawialiście się, jak komputery wyświetlają i przechowują obrazy?

***[Slajd 17]***

Ekran komputera składa się z małych punktów zwanych pikselami. Piksele to podstawowe elementy składowe obrazi cyfrowego. Liczne różnokolorowe piksele umieszczone obok siebie tworzą obraz, który możemy rozpoznać.

***[Slajd 18]***

Przyjrzyjcie się zdjęciu na ekranie. Jest obraz w skali szarości. Taki obraz zawiera tylko dwa kolory: czarny i biały, ale w różnych natężeniach. Komputery odczytują te natężenia jako liczby od 0 do 255, gdzie 0 = czarny, a 255 = biały. Tak więc obraz w skali szarości jest widziany przez komputery jako zbiór liczb. Zauważcie, że liczby te znajdują się w siatce o takim samym rozmiarze jak obraz.

Tak jest zapisywany obraz w skali szarości. A co z obrazami kolorowymi?

***[Slajd 19]***

Dlaczego akurat 255? Podstawowa jednostka instrukcji komputera to seria jedynek i zer. Jest to tak zwany system binarny.

Każdy bit reprezentuje albo zero (wył.), albo jeden (wł.).

Każdy kolor używa jednego bajta do zapisania informacji o kolorze.

Jeden bajt składa się z 8 bitów.

Mamy zatem 256 różnych możliwości i przedział od 0 do 255.

***[Slajd 20]***

Kolor jest mieszanką kolorów podstawowych: w przypadku czerwonego, zielonego i niebieskiego (Red, Green, Blue - RGB). Tak samo jak w skali szarości, również kolory są odczytywane jako liczby między 0 a 255. 0 oznacza, że piksel nie zawiera elementu koloru, a 255 oznacza, że natężenie danego koloru jest maksymalne. Wynika z tego, że zamiast jednej liczby na opisanie koloru potrzebujemy teraz 3 liczb na jeden piksel.

***[Slajd 21] - Ćwiczenie***

Zapoznajmy się bliżej z pikselami i zobaczmy, jak różne warianty liczbowe pozwalają nam utworzyć pożądany kolor. <https://www.w3schools.com/colors/colors_rgb.asp>

Na palecie kolorów znajdźcie wasz ulubiony kolor! Jaka jest wartość RGB?

Możecie stworzyć własny piksel tutaj: [www.piskelapp.com](http://www.piskelapp.com)

***[Slajdy 23–27] Wykład: Cechy obrazu [25 min]***

**Celem tej części jest prezentacja cech obrazu i sposobu ich wyodrębniania.**

***[Slajd 23]***

Na pewno wiecie, czym są puzzle! Otwórzcie tę stronę internetową z puzzlami: <https://www.jigsawexplorer.com/>

Jaką macie technikę układania puzzli? [poczekaj na odpowiedzi uczestników]

***[Slajd 24]***

Prawdopodobnie używacie jednej lub wielu z tych technik razem:

- Zaczynacie od krawędzi obrazu

- Szukacie kształtów, które znacie

- Grupujecie razem puzzle o takim samych/podobnym kolorze, np. niebo

Szukacie jakiegoś rozpoznawalnego wzoru, czegoś co, pozwoli wam znaleźć większe obiekty. Jakie są kryteria wyboru pojedynczego puzzla? Szukacie wzorów i schematów, które są **unikatowe**, **łatwe do wykrycia** i **łatwe do porównania**. Spójrzcie na to zdjęcie. Która cecha waszym zdaniem będzie tu najlepsza?

***[Slajd 25]***

Wyobraźcie sobie, że kamera bezpieczeństwa rejestruje ten obraz. U góry obrazu znajduje się sześć małych fragmentów. Waszym zadaniem jest ustalić dokładną lokalizację tych fragmentów na obrazie. Od którego zaczniecie? Który jest trudniejszy? [Poczekaj na odpowiedzi]

A i B to płaskie powierzchnie zajmujące duży obszar. Trudno jest ustalić dokładną lokalizację tych fragmentów.

C i D są dużo łatwiejsze. Są to krawędzie budynku. Możecie ustalić przybliżoną lokalizację, a trudniej będzie z lokalizacją dokładną. To dlatego, że schemat jest taki sam wzdłuż całej krawędzi.

E i F to rogi budynku. Łatwo je znaleźć. Ponieważ na rogach fragment po przesunięciu zawsze wygląda inaczej. Dlatego są to dobre cechy.

***[Slajd 26]***

Teraz przyjrzymy się jeszcze prostszemu kontekstowi. Tu są fragmenty. Jak ustalić dokładną lokalizację każdego fragmentu?

Niebieski fragment to obszar płaski i trudno go znaleźć. Gdziekolwiek przesuniecie niebieski fragment, wygląda on tak samo. Czarny fragment ma krawędź. Jeśli przesuniemy go równolegle do krawędzi, wygląda on tak samo. Czerwony fragment to róg. Gdziekolwiek go przesuniemy, wygląda on inaczej, więc jest unikatowy. Dlatego rogi uważamy za dobre cechy obrazu.

***[Slajd 27] - Czym są cechy obrazu? - Jak znaleźć cechy?***

Brawo! Teraz już wiemy, czym są cechy obrazu. Poznamy narzędzie, które pozwoli nam wyodrębnić te cechy z obrazu, abyśmy mogli je przetwarzać.

***[Slajdy 28–31] Samokształcenie: Wprowadzenie do OpenCV [90 min]***

**Celem tej części jest zaprezentowanie uczestnikom biblioteki OpenCV i jej podstawowych funkcji.**

***[Slajd 29]***

Teraz zobaczymy, jak praktyce korzystać z OpenCV. Jest to niezwykle użyteczna biblioteka, która pomoże nam w komputerowym rozpoznawaniu obrazów. Zaczniemy od aktywacji naszego środowiska wirtualnego, które utworzyliśmy na potrzeby zastosowania komputerowego rozpoznawania obrazów, a następnie otworzymy notatnik Jupyter.

***[Slajd 30]***

Zapoznajcie się z notatnikiem Jupyter. Za 30 minut omówimy zadania.

[Zarządzaj czasem zgodnie z postępami uczestników]

Brawo! Zobaczmy, jak wam poszło. Omówimy teraz zadania!

[Możesz użyć notatnika Jupyter, aby omówić zadania i wyzwania]

***[Slajd 30]***

Zróbmy sobie podsumowanie! Czego się nauczyliśmy?

- OpenCV jako użyteczna biblioteka do komputerowego rozpoznawania obrazów

- Otwieranie obrazu

- Zmiana przestrzeni barw

- Otwieranie obrazu w skali szarości

- Dostęp do pikseli

- Zmiana pikseli

- Przycinanie obrazów

- Zmiana rozmiaru obrazów

- Przetwarzanie wsadowe

Brawo! Zaczęliśmy korzystać z Open CV na potrzeby naszego rozwiązania do komputerowego rozpoznawania obrazów. Będziemy to kontynuować na etapie Doświadczenie!

***[Slajdy 32–45] Operator konwolucji [110 min]***

**Celem tej części jest prezentacja konwolucji i sposobu ich wykorzystania do edycji zdjęć i identyfikacji cech dla konwolucyjnej sieci neuronowej.**

***[Slajd 33]***

Wiemy już, że komputery zapisują obrazy w formie liczb i że piksele są zorganizowane w określony sposób, aby utworzyć obraz, który jesteśmy w stanie rozpoznać. Oznacza to, że gdy edytujemy te liczby, jednocześnie zmieniamy sposób, w jaki wyświetla się obraz. Nauczyliśmy się tego, korzystając z OpenCV.

***[Slajd 34]***

Używaliście już kiedyś edytora zdjęć? Jest to typowa funkcja w takich aplikacjach jak Instagram i wielu innych. Jakich efektów zazwyczaj używacie? Co się dzieje, gdy edytujecie obraz? [Poczekaj na odpowiedzi]

Tak, macie rację! Zmieniają się liczby w pikselach. Dziś zapoznamy się z operacją konwolucji, która zazwyczaj służy do tworzenia tych efektów.

***[Slajd 35]***

Teraz nałożymy różne efekty na obraz, który mamy w komputerze.

Użyjcie tego łącza:

http://setosa.io/ev/image-kernels/

lub http://matlabtricks.com/post-5/3x3-convolution-kernels-with-online-demo

1. Wypróbujcie efekty zaprezentowane w aplikacji internetowej

2. Utwórzcie własną konwolucję i nadajcie jej nazwę w oparciu o sposób zastosowania: wykrywanie krawędzi, kontrast, rozjaśnienie itp.

***[Slajd 36]***

Korzystając z aplikacji internetowej, wykonajcie następujące czynności:

- Zmieńcie wszystkie wartości na ujemne

- Zmieńcie wszystkie wartości na dodatnie

- Zastosujcie połączenie wartości dodatnich i ujemnych

Co zauważyliście?

***[Slajd 37]***

Poeksperymentujcie z wartościami.

- Zmieńcie 4 liczby na ujemnie. Pozostałe wartości niech będą zerowe.

- Zmieńcie jedną z nich na dodatnią.

- Obserwujcie, co się dzieje.

- Zmieńcie drugą na dodatnią.

- Obserwujcie. Zaproponujcie teorię. Teraz przetestujcie waszą teorię. Zmieńcie lokalizację czterech liczb. Powtórzcie tę czynność.

- Czy wasza teoria się sprawdza? Jeśli tak, gratulacje. Zmieńcie obraz i spróbujcie ponownie.

- Jeśli nie, poprawcie teorię. Powtarzajcie te czynności, aż osiągniecie sukces!

Omówmy to.

***[Slajd 38]***

Czas na dyskusję!

***[Slajd 39]***

Na ekranie widzimy filtr konwolucji. Jest to po prostu matryca liczb. Wyświetlamy filtr 3x3. Jak może nam to pomóc w edycji obrazu?

Wyobraźcie sobie, że umieszczamy filtr na obrazie, który chcemy edytować. Wykonacie pomnożenie elementów na filtrze i zdjęciu, dodacie liczby, podzielicie liczbę przez liczbę filtrów i rezultat będzie waszymi obrazem wyjściowym.

Później przesuniecie filtr i powtórzycie tę operację. Na końcu uzyskacie obraz wyjściowy, który wygląda jak zmodyfikowana wersja obrazu wejściowego.

***[Slajd 40]***

Co się dzieje na krawędzi obrazu? Będą tam obszary, gdzie filtry obrazu nie nachodzą na oryginalny obraz. Na przykład, lewy górny róg obrazu wejściowego ma tylko trzech sąsiadów. Można to poprawić, zwiększając wartości krawędzi o jeden na oryginalnym obrazie, nie zmieniając przy tym rozmiaru nowego obrazu.

***[Slajd 41]*** - Ćwiczenie: Operator konwolucji

Spróbujmy wykonać tę operację na papierze, żeby zrozumieć, co tu się faktycznie dzieje. Użyjemy małej matrycy reprezentującej mały obraz i zastosujemy filtr do przeprowadzenia operacji konwolucji. Sprawdźmy, jak to zadziała!

Jaki jest wasz końcowy rezultat?

Jak sobie poradziliście z rogami, gdzie już nie ma żadnych pikseli?

Są na to 2 sposoby.

- Po pierwsze, możemy zignorować wartość na rogach i wykonać konwolucję tylko na tych pikselach, na które nachodzi filtr.

- Drugi sposób to zwiększenie wartości krawędzi o jeden na oryginalnym obrazie, nie zmieniając przy tym rozmiaru nowego obrazu.

***[Slajd 42]*** - Ćwiczenie: Aplikacji internetowa do konwolucji

Teraz użyjemy komputera, aby wypróbować różne filtry i zobaczyć, jakie efekty powstaną na naszych obrazach!

Użyjcie tego łącza:

http://setosa.io/ev/image-kernels/

lub http://matlabtricks.com/post-5/3x3-convolution-kernels-with-online-demo

1. Wypróbujcie efekty zaprezentowane w aplikacji internetowej

2. Utwórzcie własną konwolucję i nadajcie jej nazwę w oparciu o sposób zastosowania: wykrywanie krawędzi, kontrast, rozjaśnienie itp.

Oczekiwane rezultaty edukacyjne:

Wykonywanie operacji konwolucji

Użyteczność operacji konwolucji

do edycji obrazu

do wykrywania cechy obrazu

***[Slajd 43]***

Oto, co należy zapamiętać z tej części naszej sesji:

Konwolucja to operacja służąca do edycji obrazu

Konwolucja jest też używana w konwolucyjnej sieci neuronowej

W konwolucyjnej sieci neuronowej konwolucja służy do uzyskiwania cech

***[Slajd 44]***

Brawo! Wiemy już, że komputery używają pikseli do zapisywania i wyświetlania obrazów. Wiemy, że operacja konwolucji służy do edycji zdjęć. Czy możecie wymienić efekty, które udało wam się zastosować?

- Tożsamość

- Wykrywanie obrazu

- Rozmycie

- itd.

***[Slajd 45]***

Brawo! Jesteśmy gotowi, aby zapoznać się z konwolucyjną siecią neuronową. Ta sieć jest popularna w komputerowym rozpoznawaniu obrazów, ponieważ efektywnie wykorzystuje konwolucje w celu zastosowania zaawansowanej sztucznej inteligencji do komputerowego rozpoznawania obrazów.

Rezultatem operacji konwolucji są cechy. Czym są cechy na obrazie? Są to obszary zainteresowania. Obszary z cieniami, krawędziami, kontrastem. Są to wzory i schematy obrazu, które pozwalają nam rozpoznać obraz.

Za chwilę dowiemy się więcej o tym, jak działa konwolucyjna sieć neuronowa!

***[Slajdy 46–71] Konwolucyjne sieci neuronowe (CNN) [120 minut]***

**Celem tej części jest zaprezentowanie sieci konwolucyjnej, czyli jednej z typowych sieci neuronowych wykorzystywanych do komputerowego rozpoznawania obrazów.**

***[Slajd 47]***

W wielu aktualnych zastosowaniach komputerowego rozpoznawania obrazów używane są zaawansowane sieci neuronowe zwane konwolucyjnymi. Teraz chcemy się dowiedzieć, jak takie sieci działają. Oto graficzne przedstawienie struktury konwolucyjnej sieci neuronowej. Jak widzicie, składa się ona z kilku warstw. Pamiętacie grę w sieć neuronową? Pamiętacie warstwy ukryte? Tu widzimy, że warstwy ukryte obejmują kilka funkcji. Zapoznamy się z nimi po kolei.

***[Slajd 48]***

Konwolucyjna sieć neuronowa składa się z 4 części. Wszystkie one współpracują, aby przetworzyć obraz i pomóc nam w zadaniach związanych z komputerowym rozpoznawaniem obrazów.

***[Slajd 49]***

Na poprzedniej sesji dowiedzieliśmy się, czym są konwolucje i jak mogą posłużyć do wyodrębnienia cech.

Co się dzieje w warstwie konwolucyjnej?

Warstwa konwolucyjna zawiera kilka filtrów, które generują poszczególne cechy. Ta warstwa umożliwia stworzenie mapy cech. Mapę cech nazywany czasem mapą aktywacji. Możemy tych terminów używać zamiennie.

Mapa cech może być zastosowana do kilku celów:

- Redukujemy rozmiar obrazu, aby efektywniej go przetwarzać.

- Koncentrujemy się na tych cechach obrazu, które przydadzą się w dalszym przetwarzaniu obrazu. Na przykład, wystarczy, że rozpoznamy czyjeś oczy, nos i usta, aby rozpoznać osobę. Nie potrzebujemy do tego całej twarzy.

***[Slajd 50]***

Następna funkcja to Rektyfikowana Jednostka Liniowa (ReLu). Niech ta nazwa was nie wystraszy. Ta funkcja po prostu eliminuje wszystkie ujemne wartości na mapie cech i pozostawia wartości dodatnie bez zmian. Tym sposobem wprowadzana jest nieliniowość. Co to oznacza? Spójrzcie na te dwa wykresy: ten po lewej to wykres liniowy, ten po prawej to wykres ReLu. Jaka jest różnica? Wykresy ReLu zaczynają się od poziomej linii prostej i liniowo wskazują osiągnięcie wartości dodatniej. Do czego jest nam to potrzebne? Chcemy, by zmiana koloru była bardzo wyraźnie widoczna. Jest to wzmocnienie warstwy aktywacji.

***[Slajd 51]***

Spójrzcie na ten obraz. Oryginalny obraz najpierw trafia do warstwy konwolucji. Tu widać rezultat operacji konwolucji. Następną operacją jest ReLu. Na obrazie po konwolucji jest płynne szare przejście od czarnego do białego. Po zastosowaniu funkcji ReLu ta zmiana jest wyraźna i skokowa.

***[Slajd 52]***

Podpróbkowanie nazywane jest też próbkowaniem w dół. Chcemy w tym przypadku zmniejszyć rozmiar obrazu, zachowując jego ważne cechy. Pamiętacie, dlaczego chcemy to zrobić? Tak, chcemy zapewnić, że nasze dane będą efektywnie przetwarzane.

To jest przykład maksymalnego próbkowania, gdy wybieramy największy element w określonej przestrzeni (np. 2x2).

Są też inne wersje próbkowania, np. średnie albo sumaryczne próbkowanie.

***[Slajd 53]***

Próbkowanie pozwala nam:

- Zmniejszyć obraz i ułatwić jego obróbkę

- Zredukować liczbę parametrów i kontrolować przetrenowanie

- Uodpornić obraz na drobne transformacje, zniekształcenia i zmiany na obrazie wejściowym Oznacza to, że niewielka różnica na obrazie wejściowym wygeneruje bardzo podobny obraz próbkowany.

- Tworzyć przedstawienia obrazu/stałego schematu, które umożliwią nam wykrywanie obiektów na obrazie niezależnie od ich lokalizacji.

***[Slajd 54]***

Próbkowanie uodparnia obraz na drobne transformacje, zniekształcenia i zmiany na obrazie wejściowym Oznacza to, że niewielka różnica na obrazie wejściowym wygeneruje bardzo podobny obraz próbkowany.

***[Slajd 55]***

Pamiętajcie, że choć ludzkie oko może nie wiedzieć tych obrazów wyraźnie, to komputer ich używa ich do przetwarzania obrazu (zbioru liczb).

***[Slajd 56]***

Na jakim jesteśmy etapie? Wiemy, jak działają sieci konwolucyjne, funkcja ReLu i próbkowanie w dół. Czy mamy 3 ochotników, którzy przypomną nam, jak działają te warstwy i jakie są ich funkcje?

***[Slajd 57]***

Warstwa konwolucyjna, funkcja ReLu i warstwa próbkowania pojawią się też przy okazji konwolucyjnych sieci neuronowych.

***[Slajd 58]***

Czego się dowiedzieliśmy do tej pory? Poznaliśmy procesy redukcji rozmiaru obrazu i wzmocnienia jego cech. Co dalej?

***[Slajd 59]***

Dotarliśmy do warstwy w pełni połączonej. Nie jest to dla nas nic nowego. To ta sama warstwa, o której mówiliśmy przy okazji sieci neuronowych. „W pełni połączona” oznacza, że każdy neuron z poprzedniej warstwy jest połączony z każdym neuronem z następnej warstwy.

Poprzednio rezultat sieci konwolucyjnej, funkcji ReLu i próbkowania w dół generował cechy obrazu wysokiego poziomu z warstwy wejściowej. Celem warstwy w pełni połączonej jest użycie tych cech do klasyfikacji obrazu wejściowego według różnych klas w oparciu o zbiór danych do trenowania. Każdy neuron otrzyma różne funkcje i wagę, aby dać nam współczynnik prawdopodobieństwa, że obraz zawiera określony obiekt.

***[Slajd 60]***

Warstwa w pełni połączona generuje współczynnik prawdopodobieństwa w oparciu o obraz wejściowy i dane do trenowania.

***[Slajd 61]***

Tak jest przeprowadzane wnioskowane za pomocą konwolucyjnej sieci neuronowej. Gratulacje! Teraz już wiecie, jak działają konwolucyjne sieci neuronowe!

***[Slajd 62]*** - Ćwiczenie: Testowanie sieci konwolucyjnej za pomocą zbioru danych MNIST

Zobaczmy, jak to wygląda w praktyce. Zobaczcie to demo i sprawdźcie, czy rozumiecie, jak różne warstwy funkcjonują w procesie klasyfikacji liczb.

***[Slajd 63]*** - Ćwiczenie: Testowanie sieci konwolucyjnej za pomocą zbioru danych MNIST

Dowiedzieliście się, jak konwolucyjne sieci neuronowe umożliwiają klasyfikację obrazów i lokalizację. Aby to było możliwe, konwolucyjna sieć neuronowa (CNN) musi być wcześniej wytrenowana. Przyszedł czas na pytanie: Jak trenować CNN?

Mówiąc najprościej, trenowanie wygląda następująco:

1. Wprowadzamy dane trenowania do maszyny. Dane posiadają etykiety. Są sto dane „poprawne”.

2. Dostarczamy dane wejściowe (czyli dane testowe) i przetwarzamy je za pomocą konwolucyjnej sieci neuronowej. Uzyskujemy określoną predykcję.

3. Maszyna porównuje predykcje z sieci neuronowej z danymi trenowania. Wystąpi niezgodność między nowymi danymi a danymi ze zbioru do trenowania. Nazwiemy tę niezgodność błędem.

4. Maszyna zmieni funkcję w sieci neuronowej, aby zredukować błąd. To jest powtarzane, aż współczynniki prawdopodobieństwa będą wystarczająco dokładne.

Przejdźmy do szczegółów.

***[Slajd 64]***

Pierwszy krok do dostarczenie danych trenowania z etykietami. Co to oznacza? [Poczekaj na odpowiedzi] Dane z etykietami oznaczają, ze zbiór obrazu jest poprawnie nazwany.

***[Slajd 65]***

Obraz wejściowy jest przetwarzamy w sieci. Zostanie porównany z danymi trenowania. Na początku współczynnik prawdopodobieństwa będzie niezgodny z danymi trenowania. Wystąpi duży błąd.

***[Slajd 66]***

Wygenerowana wartość błędu zostanie użyta do zmiany parametrów CNN. Parametry zostaną zmienione, a błąd będzie mniejszy.

***[Slajd 67]***

Ten proces zostanie powtórzony. Parametry będą zmieniane, aż współczynnik prawdopodobieństwa będzie wystarczająco dokładny do naszych celów.

***[Slajd 68]***

Podsumujmy cały proces. Czy ktoś chce to zrobić na ochotnika?

***[Slajd 69]*** - Ćwiczenie: Maszyna zdolna do nauki

Wypróbujcie aplikację internetową z maszyną zdolną do nauki!

***[Slajd 70]*** - Ćwiczenie: Działanie CNN w praktyce

Zobaczmy, co może zdziałać wytrenowany model. Przejdźcie na stronę https://cloud.google.com/vision/ i wypróbujcie interfejs API do rozpoznawania obrazów!

***[Slajd 71]*** - Quiz

Brawo dla wszystkich! Zróbmy quiz, który sprawdzi, czy dobrze wszystko rozumiecie! Sprawdźmy, co zapamiętaliście!

<https://create.kahoot.it/share/computer-vision-convolutional-neural-network/2b8357a8-0147-4301-939e-df6286712431>

***[Slajdy 72–76] Maszyna wektorów nośnych (SVM [30 min]***

**Celem tej części jest prezentacja maszyny wektorów nośnych i zastosowania tego algorytmu do uczenia maszynowego.**

***[Slajd 72]***

Wiemy już, czym jest konwolucyjna sieć neuronowa (CNN), często używana do komputerowego rozpoznawania obrazów.

Istnieją też prostsze modele służącego do tych samych celów. Na etapie pozyskiwania danych poznaliśmy algorytm K najbliższych sąsiadów. Dziś poznamy jeszcze jeden algorytm.

***[Slajd 73]***

Maszyna wektorów nośnych to popularny algorytm uczenia maszynowego, który można stosować do problemów związanych z klasyfikacją. Ten algorytm powstał jeszcze przed sieciami neuronowymi i był używany przez naukowców i inżynierów do rozwiązywania problemów w dziedzinie komputerowego rozpoznawania obrazów, Choć kuszące jest stosowanie sieci neuronowych do wszystkich problemów, to w niektórych przypadkach prostsze algorytmy (np. maszyna wektorów nośnych) działają szybciej i są dokładniejsze. Warto nauczyć się technik uczenia maszynowego i korzystać z nich, gdy pozwala na to problem, którym się zajmujemy.

***[Slajd 74]***

Czym jest zatem maszyna wektorów nośnych? Zobaczmy to na przykładzie problemu klasyfikacyjnego z 2 klasami. Na ekranie widzimy zbiory danych z różnych klas, przedstawione za pomocą dwóch kolorów. Jak oddzielić od siebie te dwie grupy? Można narysować linię między punktami i stworzyć potencjalną granicę podziału. Spójrzcie na różne linie. Która jest najlepsza do wprowadzenia klasyfikacji?

W maszynie wektorów nośnych trenowanie modelu oznacza poszukiwanie linii najlepiej oddzielających 2 klasy.

***[Slajd 75]***

Jaka działa maszyna wektorów nośnych? W przypadku danych 2D znajduje linię najlepiej rozdzielającą zbiór danych na różne grupy. Ta linia nazywana jest granicą decyzyjną.

Dane 2D będą mieć granicę decyzyjną w postaci linii (1D). Dane 3D będą mieć granicę decyzyjną w postaci płaszczyzny (2D). W przypadku danych większych niż 3D nie będziemy w stanie zwizualizować danych i granicy decyzyjnej, ale granica decyzyjna będzie nazywana hiperpłaszczyzną.

***[Slajd 76]***

Jeszcze kilka słów o maszynie wektorów nośnych. Punkty najbliżej granicy decyzyjnej to wektor nośny. Wektor nośny rozwiązuje problem optymalizacji:

1. Wektory nośne znajdują się najdalej od granicy decyzyjnej.

2. Dwie klasy leżą po przeciwnym stronach granicy decyzyjnej.

Przećwiczymy to w praktyce podczas etapu Doświadczenie!

### 

### ***[Slajdy 77–80] Dyskusja na temat etyki [40 min]***

**Celem tej części jest omówienie potencjalnego ryzyka i niewłaściwego zastosowania zaawansowanej sztucznej inteligencji w komputerowym rozpoznawaniu obrazów.**

***[Slajd 78]***

Dziś dowiedzieliśmy się, jak komputery umożliwiają rozpoznawanie, lokalizację i modyfikację obrazów. Pamiętacie, jak kopiowaliśmy obrazy za pomocą OpenCV? Sztuczna inteligencja też to potrafi i to znacznie lepiej!

Co się stanie, gdy popadniemy w skrajność? Co się stanie, gdy zmodyfikujemy obraz w taki sposób, że wygląda jak prawdziwy?

***[Slajd 79]***

Obejrzyjcie ten film opowiadający o tym, czym jest deepfake.

<https://www.youtube.com/watch?v=AmUC4m6w1wo>

***[Slajd 80]***

Pytania do refleksji:

- W jaki sposób ogromna liczba deepfakes wpłynie na nasze zaufanie do mediów?

- Jak można odróżnić deepfakes od prawdziwych obrazów?

- Co możecie zrobić jako odpowiedzialni użytkownicy AI, aby ograniczyć niewłaściwe wykorzystanie AI?

[Po dyskusji podsumuj sesję, przypominając najważniejsze rzeczy do zapamiętania]

## **4. Wskazówki dotyczące rozwiązywania problemów - WAŻNE**

### ***Typowe błędy/problemy sprzętowe***

|  |  |  |  |
| --- | --- | --- | --- |
|  | **BŁĘDY/PROBLEMY** | **MOŻLIWE PRZYCZYNY** | **ROZWIĄZANIE** |
| 1 | Kod w notatniku Jupyter nie działa | Oprogramowanie nie zostało zainstalowane/jest nieprawidłowo skonfigurowane | Ponownie zainstaluj oprogramowanie zgodnie z instrukcją w części 2. |

### 

### 

### **5. Rubryka oceny i tryb oceny**

### 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Umiejętności** | **Tryb** | **Słabe** | **Średnie** | **Dobre** |
| Wyjaśnienie sposobu analizy obrazów przez komputery | Quiz | Uczestnicy nie potrafią opisać, jak komputery analizują obrazy | Uczestnicy potrafią opisać, jak komputery analizują obrazy, jeśli otrzymają podpowiedzi | Uczestnicy potrafią opisać, jak komputery analizują obrazy  bez żadnych podpowiedzi |
| Wyjaśnienie działania konwolucji | Zadanie praktyczne | Uczestnicy nie potrafią przeprowadzić operacji konwolucji | Uczestnicy potrafią przeprowadzić operacji konwolucji, jeśli otrzymają podpowiedzi | Uczestnicy nie potrafią przeprowadzić operacji konwolucji bez podpowiedzi |
| Opisanie sposobu działania konwolucyjnych sieci neuronowych (CNN) | Quiz | Uczestnicy nie potrafią opisać, jak działają CNN | Uczestnicy potrafią opisać, jak działają CNN, jeśli otrzymają podpowiedzi | Uczestnicy potrafią opisać, jak działają CNN bez podpowiedzi |
| Opisanie sposobu działania maszyny wektorów nośnych (SVM) | Dyskusja | Uczestnicy nie potrafią opisać, jak działa SVM | Uczestnicy potrafią opisać, jak działa SVM, jeśli otrzymają podpowiedzi | Uczestnicy potrafią opisać, jak działa SVM bez podpowiedzi |
| Wskazanie zastosowań komputerowego rozpoznawania obrazów (CV) | Quiz | Uczestnicy nie potrafią wymienić zastosowań CV | Uczestnicy potrafią wymienić zastosowania CV omówione na warsztatach | Uczestnicy potrafią wymienić zastosowania CV omówione na warsztatach oraz inne zastosowania |

### 

### **6. Polecane lektury**

* Wprowadzenie do komputerowego rozpoznawania obrazów <https://www.oreilly.com/library/view/practical-computer-vision/9781449337865/ch01.html#_why_learn_computer_vision>
* Zrozumienie cech <https://docs.opencv.org/3.4/db/d27/tutorial_py_table_of_contents_feature2d.html>
* Jak łatwo automatyzować nadzór za pomocą uczenia głębokiego <https://medium.com/nanonets/how-to-automate-surveillance-easily-with-deep-learning-4eb4fa0cd68d>
* Intuicyjne objaśnienie konwolucyjnych sieci neuronowych <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>
* Kompleksowy przewodnik dotyczący konwolucyjnych sieci neuronowych (CNN) <https://www.superdatascience.com/blogs/the-ultimate-guide-to-convolutional-neural-networks-cnn>
* Zrozumienie algorytmu maszyny wektorów nośnych na podstawie przykładów (wraz z kodem) <https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>